



## ISD Checklist

**Number:** 580-CK-008-01  
**Effective Date:** August 1, 2004  
**Expiration Date:** August 1, 2009

**Approved By:** (signature)  
**Name:** Joe Hennessy  
**Title:** Chief, ISD

**Asset Owner:** 580/Information Systems Division (ISD)  
**Title:** Contents of the Software Critical Design Review

**Asset Type:** Checklist  
**PAL Number:** 2.3.2.5

### Contents of the Software Critical Design Review (CDR)

*The purpose of this review is to determine whether the detailed design satisfies requirements and is ready to be implemented in code and tested. Use the following checklist to ensure that key elements of the design are presented for review, along with supporting technical, management, process, and status information.*

- ☐ **Introduction**
  - ☐ Goals of this review; review prerequisites, scope (e.g., all subsystems, COTS/GOTS, database schemas, and firmware), agenda, and products
  - ☐ Introduction to the review panel and review process to be followed (e.g., Request for Action (RFA) or Review Item Disposition (RID) reports, schedule)
- ☐ **Project Overview**
  - ☐ Purpose and overview of the software project or overall Mission Project (e.g., Mission goals, orbit, launch, operational characteristics)
  - ☐ Software project organization and key personnel, including (if applicable) how the project fits within the overall Mission organization and identification of the project's systems engineers at the Mission Project level
- ☐ **Management Overview**, highlighting changes since Software Preliminary Design Review (PDR)
  - ☐ Management activities, including regular management and technical meetings, status reporting
  - ☐ V&V plan, including milestone and peer reviews, walkthroughs, and external reviews
  - ☐ Status of Requests for Action (RFAs) or Review Item Dispositions (RIDs) and responses from the Software PDR and, if applicable, the Project PDR and/or CDR
  - ☐ Status of ICDs/IRDs and other external dependencies (documents, software, hardware, etc.)
  - ☐ Documentation plan, including each document's status and when it will be baselined
  - ☐ Product Assurance and Software Safety plans and activities
  - ☐ Independent verification and validation (IV&V) plans and status
  - ☐ Collection and analysis of software project metrics
  - ☐ Collection and reporting of Lessons Learned; infusion of Lessons Learned in earlier phases and/or projects
- ☐ **Development Process and Plans**, highlighting changes since PDR
  - ☐ Software requirements definition and management process, including documents used and produced, V&V, and baselines
  - ☐ Design process, including methodology and standards used, design documentation produced, inspections and reviews
  - ☐ Implementation process, incl. standards, review process, problem reporting, unit test, integration
  - ☐ Build plan
  - ☐ Configuration Management (CM) processes, including discrepancy reporting and tracking (development and post-release)
- ☐ **Software Overview**, highlighting changes since PDR
  - ☐ Overview of functional requirements and operations concepts
  - ☐ System (software and hardware) architecture, external interfaces and end-to-end data flow
  - ☐ Software context diagram showing each subsystem or major component
  - ☐ Failure detection and correction (FDC) requirements, approach, and detailed design
  - ☐ IT Security Requirements (Mission-specific)
  - ☐ Software Requirements Verification Matrix (mapping requirements to subsystems/components)
  - ☐ Development environment (e.g., hardware diagram, operating system(s), compilers, DBMS, tools)

## Contents of the Software Critical Design Review (CDR) (Continued)

**For each subsystem or major component,**

- ❑ **Requirements**
  - ❑ Functional and initialization requirements allocated to the subsystem/component, with any safety-critical and IT Security requirements highlighted
  - ❑ Requirements changes since PDR
- ❑ **Detailed Design**
  - ❑ Design changes since PDR
  - ❑ Reused/heritage software or functionality from previous projects; necessary modifications
  - ❑ Subsystem/component context diagram
  - ❑ System design diagram (e.g., Level 0 data flow diagram or UML)
    - For each task in the system design diagram*
      - ❑ Design diagrams for the task
      - ❑ Description of functionality and operational modes
      - ❑ Resource and utilization constraints (e.g., CPU, memory); how the software will adapt to changing margin constraints; performance estimates
      - ❑ Data storage concepts and structures
  - ❑ Identification and formats of input and output data
  - ❑ Interrupts and/or exception handling, including event, FDC, and error messages
  - ❑ IT Security features
  - ❑ Current status and issues
- ❑ **Software Testing**
  - ❑ Test team roles, functions, support required, and charter
  - ❑ Documentation – titles and status of test plans, procedures, and traceability matrices
  - ❑ Test levels (e.g., unit testing, integration testing, system testing) – description, who executes, test environment, standards followed, verification methodologies
  - ❑ Testing preparation and execution activities, incl. testing of reused/heritage software if applicable
  - ❑ Build test timeline and ordered list of components and requirements to be tested in each build
  - ❑ Test environments for each test level – diagram and description of tools, testbeds, facilities
  - ❑ Software requirement verification recording, monitoring, and current status – databases and test reports; sample test verification matrix
  - ❑ Mission Project V&V plans and status
  - ❑ System and acceptance testing – operational scenarios to be tested, including stress tests and recovery testing, if applicable
  - ❑ Acceptance process – reviews (e.g., Acceptance Test Readiness Review, Acceptance Test Review), approval, and signoff processes
- ❑ **Delivery, Installation, and Maintenance**
  - ❑ Disposition of source code and tools, handling of load images, installation of databases, etc.
  - ❑ Version identification and documentation
  - ❑ Maintenance plan, if applicable, including disposition of COTS components (source code, licenses, etc.)
- ❑ **Software status**
  - ❑ Detailed schedule timeline, showing phases, milestones, and current status
  - ❑ Current software size estimate; staffing and cost/effort status
- ❑ **Risks** – with categories, frequency of review, consequences, and risk mitigation strategies
- ❑ **Issues, TBDs, and action items**